



Two approximation methods to synthesize the power spectrum of fractional Gaussian noise[☆]

Sergio Ledesma^{a,*}, Derong Liu^b, Donato Hernández^a

^a*FIMEE School Of Engineering, University of Guanajuato, 912 Tampico, Salamanca, Gto. 36730, México*

^b*Department of Electrical and Computer Engineering, University of Illinois, 851 S. Morgan Street (MC 154), Chicago, IL 60607, USA*

Available online 1 March 2007

Abstract

The simplest models with long-range dependence (LRD) are self-similar processes. Self-similar processes have been formally considered for modeling packet traffic in communication networks. The fractional Gaussian noise (FGN) is a proper example of exactly self-similar processes. Several numeric approximation methods are considered and reviewed, two methods are found that are able to provide a better accuracy and less running time than previous approximation methods for synthesizing the power spectrum of FGN. The first method is based on a second-order approximation. It is demonstrated that a parabolic curve can be indirectly used to approximate the power spectrum of FGN. The second method is based on cubic splines. Despite the fact that splines cannot be used directly to approximate the power spectrum of FGN, they can, however, considerably simplify the calculations while maintaining high accuracy. Both of the methods proposed can be used to estimate the Hurst parameter using Whittle's estimator. Additionally, they can be used on synthesis of LRD sequences.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Self-similarity; Long-range dependence; Fractional Gaussian noise; Power spectrum synthesis; Hurst parameter

1. Introduction

A standard assumption of time series analysis is that observations separated by a large time span are roughly independent. However, for long-range dependence (LRD), these observations are not independent, see Beran (1986, 1992a, b, 1994) and Fox and Taqqu (1986). The Hurst parameter, denoted by H , is the measure of LRD. Larger values of H mean greater LRD. Two methods are proposed to approximate the power spectrum of fractional Gaussian noise (FGN) (a self-similar process). These methods can be used to synthesize self-similar series and estimate the value of H for a time series. Because both methods are faster and more accurate than previous methods, they are ideal when a quick value of H using Whittle's estimator is required.

Traffic measurements obtained from packet networks have convincingly established the existence of statistical features that are characteristic of fractal traffic processes (Adas and Mukherjee, 1995; Erramilli et al., 1996; Jeong, 2002), particularly in the sense that these features regularly span many time scales, see Erramilli et al. (1994) and Taqqu

[☆] The synthetic LRD sequences generated using the Spline method are given in the supplementary material.

* Corresponding author. Department of Electrical Engineering and Computer Science, University of Illinois at Chicago, 851 S. Morgan Street (MC 154), Chicago, IL 60607, USA. Tel.: +1 312 359 4475.

E-mail address: selo@salamanca.ugto.mx (S. Ledesma).

and Teverovsky (1997). It has been shown that the concept of self-similarity can be applied to WAN and LAN traffic, see Leland et al. (1994), Paxson and Floyd (1995) and Willinger et al. (1997).

This paper is organized as follows. In Section 2, we present background information on LRD and self-similarity. In Section 3, we present a brief description of existing methods for approximating the power spectrum of FGN. In Section 4, we develop two methods for reducing the computational complexity of the power spectrum of FGN. These two methods can be used to speedily generate self-similar time series and calculate Whittle's estimator. In Section 5, we perform a statistical analysis to compare existing methods with the two methods developed in this paper. Finally, Section 6 presents conclusions and direction for future work.

2. Background information

Consider a stochastic process $X = \{X_t; t = 0, 1, 2, \dots\}$. For each $m = 1, 2, 3, \dots$, let

$$X^{(m)} = \{X_k^{(m)}; k = 1, 2, 3, \dots\},$$

denote a new time series obtained by averaging the original series X ; that is,

$$X_k^{(m)} = \frac{X_{km-m} + \dots + X_{km-1}}{m}.$$

The new sequence $X_k^{(m)}$ is frequently used to estimate the Hurst parameter, see Roughan et al. (1998). Generally, the Hurst parameter can be estimated using the:

- Analysis of the variances of the aggregate processes $X^{(m)}$.
- Time-domain analysis based on the re-scaled adjusted range (R/S-statistics).
- Frequency-domain analysis based on the periodogram or Whittle's estimator.
- Wavelet-based estimator, see Veitch and Abry (1998, 1999)

The most straightforward models with LRD are the self-similar processes, see Cox (1984). X is exactly or asymptotically second-order self-similar if the corresponding aggregate processes $X^{(m)}$ are the same as X or become indistinguishable from X , at least with respect to their autocorrelation functions, see Leland et al. (1994), Beran et al. (1995), Paxson and Floyd (1995) and Willinger et al. (1995). Self-similarity and second-order self-similarity are equivalent concepts for Gaussian process since their finite dimensional distributions are, by definition, Gaussian. Fractional Gaussian noise (FGN) is an example of exactly self-similar processes, and fractional autoregressive integrated moving average process is an example of asymptotically self-similar processes.

Paxson (1997) proposed an analytical method for synthesizing FGN using the results of Flandrin (1992). This ingenious method makes the assumption that if the power spectrum of FGN is given, then it is possible to easily create a frequency-domain sample path. However, long memory processes are characterized by a nonfinite spectrum at the zero frequency, and in practice, the frequency values must be chosen so that none of them takes the value of zero, i.e. $\pi/u, 2\pi/u, \dots, \pi$ where $u > 1$ is an integer. Thus, using the fast Fourier transform (FFT) algorithm, it is feasible to utilize this frequency-domain sample path to synthesize a time-domain sample sequence for FGN.

Following the notation of Beran (1986) for an FGN process, and letting λ be the discrete frequency, the power spectrum for $0 < H < 1$ and $-\pi \leq \lambda \leq \pi$ is

$$f(\lambda; H) = (1 - \cos \lambda) \Upsilon(H) [|\lambda|^{-2H-1} + B(\lambda; H)], \tag{1}$$

where

$$\Upsilon(H) = 2 \sin(\pi H) \Gamma(2H + 1)$$

and

$$B(\lambda; H) = \sum_{k=1}^{\infty} [(2\pi k + \lambda)^{-2H-1} + (2\pi k - \lambda)^{-2H-1}]. \tag{2}$$

As clearly indicated by Paxson (1997), the primary problem of using (1) to accurately calculate the power spectrum of FGN is the infinite summation for $B(\lambda; H)$. Eq. (2) establishes that for a given value of H and each value of λ there is a unique value of $B(\lambda; H)$ that can be calculated through the evaluation of an infinite summation. For LRD sequence synthesis, this implies that the number of values of λ that Eq. (2) must be evaluated directly depends on the synthetic sequence length, making critical to reduce the computation of this infinite summation. Unfortunately, a close form for (2) is unknown. There are, however, several methods for approximating this equation. These methods significantly reduce the computation of the power spectrum of FGN.

Alternatively, parametric methods can be used to generate LRD sequences. For a comparison with existing parametric methods to generate LRD series, applications in network simulation and detailed information about how to generate a time series using the power spectrum of FGN see Paxson (1997).

In Section 3, we will briefly review several methods to approximate (2). To simplify the notation and to present a consistent format for all these methods, we define

$$B_{M:N}(\lambda; H) = \sum_{k=M}^N [(2\pi k + \lambda)^{-2H-1} + (2\pi k - \lambda)^{-2H-1}]. \quad (3)$$

This equation is somehow similar to Eq. (2); the main differences between Eqs. (2) and (3) are the values of the summation indexes. While (2) is an infinite summation, (3) has a finite number of terms. It can be easily seen that Eq. (2) can be expressed as $\lim_{N \rightarrow \infty} B_{1:N}(\lambda; H)$. Eq. (3) provides a convenient way to analytically manipulate the bound indexes of the summation; i.e. the summation of the second and third terms of Eq. (2) can be expressed as $B_{2:3}(\lambda; H)$.

To speed up the calculation of the power spectrum of FGN, it is required to approximate the infinite summation of (2). Several methods have been proposed to approximate this equation. This paper will review all these methods and introduce two new ones.

3. Related work

We will briefly review existing methods for approximating (2) that can be used for LRD series synthesis and to calculate Whittle's estimator. There are three methods to approximate this equation. Because all of these approaches are pretty fast and accurate, they impose tough requirements on the design of new approximation methods.

3.1. Paxson approximation

Paxson (1997) proposed an approximation method to notable speed up the heavy computation of (2). His method is based on the fact that Eq. (2) is bounded. The approach used by Paxson, 1997 is given as follows:

$$B_{\text{Paxson}}(\lambda; H) = B_{1:3}(\lambda; H) + B_{4:\infty}(\lambda; H). \quad (4)$$

That is, Paxson's approximation, denoted by $B_{\text{Paxson}}(\lambda; H)$, keeps the first three terms of (2), and approximate the remaining ones as shown

$$B_{4:\infty}(\lambda; H) \approx \frac{1}{8H\pi} \sum_{k=3}^4 [(2\pi k + \lambda)^\eta + (2\pi k - \lambda)^\eta], \quad (5)$$

where $\eta = -2H$.

3.2. First-order approximation

Following the notation of (3), Ledesma and Liu (2000) showed that it is possible to get an accurate approximation by keeping the first two terms in Eq. (2). Thus, Eq. (2) can be approximated by

$$B_{\text{FOA}}(\lambda; H) = B_{1:2}(\lambda; H) + p\lambda + q, \quad (6)$$

where $B_{\text{FOA}}(\lambda; H)$ denotes the first-order approximation. Eq. (6) indicates that the first-order approximation (FOA) method uses a first-order function of λ , $f_{\text{FOA}}(\lambda) = p\lambda + q$, to approximate $B_{3:\infty}(\lambda; H)$. The values of p and q are calculated to minimize the mean-squared error defined by

$$E_{\text{FOA}} = \int_0^\pi [B_{3:\infty}(\lambda; H) - (p\lambda + q)]^2 d\lambda. \tag{7}$$

The minimum value of E_{FOA} is obtained by setting

$$\partial E_{\text{FOA}} / \partial p = 0 \quad \text{and} \quad \partial E_{\text{FOA}} / \partial q = 0.$$

Using Leibnitz's rule to derive under the integral sign (Spiegel and Liu, 1998; Ledesma and Liu, 2000) we obtain

$$p = -\frac{6}{\pi^2} F(H) + \frac{12}{\pi^3} G(H) \tag{8}$$

and

$$q = \frac{4}{\pi} F(H) - \frac{6}{\pi^2} G(H), \tag{9}$$

where $F(H)$ for $0.5 \leq H < 1$ is defined by

$$F(H) = \sum_{k=3}^{\infty} \left[\frac{(2\pi k - \pi)^{-2H} - (2\pi k + \pi)^{-2H}}{2H} \right] \tag{10}$$

and $G(H)$ for $0.5 < H < 1$ by

$$G(H) = \sum_{k=3}^{\infty} \left[\frac{(2\pi k + \pi)^{-2H+1} + (2\pi k - \pi)^{-2H+1} - 2(2\pi k)^{-2H+1}}{-2H + 1} \right] + \sum_{k=3}^{\infty} \left[\frac{2\pi k(2\pi k + \pi)^{-2H} + 2\pi k(2\pi k - \pi)^{-2H} - 2(2\pi k)^{-2H+1}}{2H} \right]. \tag{11}$$

An expression for $G(H)$ when $H = 0.5$ was found by Ledesma and Liu (2000).

Despite the complexity of (10) and (11), the FOA method is very efficient due to the fact that the computation of $F(H)$ and $G(H)$ is not a function of λ . That is, given a value of H , it is possible to find $F(H)$ and $G(H)$, to calculate p and q using (8) and (9). Thus, $B_{3:\infty}(\lambda; H)$ can be approximated using $p\lambda + q$ instead of the direct calculation of the infinity summation in (3) when $M = 3$ and $N \rightarrow \infty$.

3.3. Euler Maclaurin-based approximation

An excellent method for approximating (2) was presented by Purczynsky and Wlodarski (2006). This novel method is based on the Euler–Maclaurin summation formula as clearly shown in (12). See Dahlquist and Bjorck (1974) and Kreyszig (1993) for relevant information about polynomial approximation. Throughout this paper, we will call this method the Euler–Maclaurin-based approximation (EM), described by

$$B_{\text{EM}}(\lambda; H) = \frac{c_1(c_2 + \lambda)^\eta + c_1(c_2 - \lambda)^\eta}{H} + c_3[(c_4 + \lambda)^{\eta-1} + (c_4 - \lambda)^{\eta-1}], \tag{12}$$

where $\eta = -2H$ as previously defined, and $B_{\text{EM}}(\lambda; H)$ denotes the approximated value of $B(\lambda; H)$ when the Euler–Maclaurin-based approximation is used. Table 1 shows the coefficients, c_1 , c_2 , c_3 , and c_4 ; they were calculated to minimize the mean-squared error and the mean absolute error.

Table 1
Values for the c -coefficients for the EM method

Hurst parameter	c_1	c_2	c_3	c_4
$0.50 \leq H \leq 0.65$	0.07743	8.2466	0.87375	6.1790
$0.65 \leq H < 1$	0.07612	8.1650	0.87880	6.1900

4. Two new methods

In this section, we develop two methods for approximating (2). Both approaches considerably reduce the computational complexity and drastically improve the accuracy when compared with existing methods. It is important to note that existing approaches are very fast and accurate, making extremely difficult for logarithmic, exponentials, high-order polynomials or other kind of approximations to be considered for designing new methods to approximate Eq. (2).

4.1. Second-order approximation

In order to be able to use a second-order approximation (SOA) and still get a smaller running time than the FOA method, it is required to retain as few terms as possible from the original series. It will be shown that a SOA can be used to calculate the summation of (2) by keeping only the first term of the infinite summation as

$$B_{SOA}(\lambda; H) = B_{1:1}(\lambda; H) + m\lambda^2 + p\lambda + q, \tag{13}$$

where $B_{SOA}(\lambda; H)$ denotes the approximated value of $B(\lambda; H)$ when utilizing the SOA method.

Eq. (13) implies that an accurate value for $B_{2:\infty}(\lambda; H)$ can be found by direct evaluation of $m\lambda^2 + p\lambda + q$ with appropriate values of m, p , and q . In order to accurately estimate these values, it is necessary to conveniently define the error of the SOA method as

$$E_{SOA} = \int_0^\pi [B_{2:\infty}(\lambda; H) - (m\lambda^2 + p\lambda + q)]^2 d\lambda. \tag{14}$$

The coefficients m, p , and q are calculated to minimize the mean-squared error in the interval $0 \leq \lambda \leq \pi$, and their optimum values can be found when

$$\frac{\partial E_{SOA}}{\partial m} = 0, \quad \frac{\partial E_{SOA}}{\partial p} = 0 \quad \text{and} \quad \frac{\partial E_{SOA}}{\partial q} = 0. \tag{15}$$

Applying Leibnitz's rule to (15), using (14), and after some algebra (see Ledesma and Liu, 2000), we get a system of equations for m, p and q as shown:

$$\begin{aligned} \frac{\pi^3}{3}m + \frac{\pi^2}{2}p + \pi q &= \phi(H), \\ \frac{\pi^4}{4}m + \frac{\pi^3}{3}p + \frac{\pi^2}{2}q &= \gamma(H), \\ \frac{\pi^5}{5}m + \frac{\pi^4}{4}p + \frac{\pi^3}{3}q &= \zeta(H), \end{aligned} \tag{16}$$

where

$$\phi(H) = \frac{\pi^\eta}{\eta} \sum_{k=2}^\infty (\rho_k^\eta - \beta_k^\eta), \tag{17}$$

$$\gamma(H) = \frac{\pi^\mu}{\mu} \sum_{k=2}^\infty (\rho_k^\mu + \beta_k^\mu) + \frac{\pi^\mu}{\eta} \sum_{k=2}^\infty \left[\frac{2\alpha_k^\mu}{\mu} - \alpha_k(\rho_k^\eta + \beta_k^\eta) \right] \tag{18}$$

Table 2
Second-order approximation coefficients for the SOA method

H	m	p	q
0.6	2.5828×10^{-4}	-6.8487×10^{-5}	1.7227×10^{-2}
0.7	1.7925×10^{-4}	-5.2019×10^{-5}	9.3262×10^{-3}
0.8	1.2351×10^{-4}	-3.9006×10^{-5}	5.1490×10^{-3}
0.9	8.4589×10^{-5}	-2.8920×10^{-5}	2.8852×10^{-3}

and

$$\zeta(H) = \pi^{\eta+2} \sum_{k=2}^{\infty} \left[\frac{(\rho_k^{\eta+2} - \beta_k^{\eta+2})}{\eta + 2} + \frac{\alpha_k^2(\rho_k^{\eta} - \beta_k^{\eta})}{\eta} - \frac{2\alpha_k(\rho_k^{\mu} - \beta_k^{\mu})}{\mu} \right], \tag{19}$$

with $\mu = \eta + 1$, $\alpha_k = 2k$, $\beta_k = 2k - 1$ and $\rho_k = 2k + 1$. Eq. (17) is valid for $0.5 \leq H < 1$. Eqs. (18) and (19) are valid only for $0.5 < H < 1$; the short range dependence case, $H = 0.5$ is discussed in Appendix A. The convergence of the infinite series of (17)–(19) and (19) is discussed in Appendix B.

Solving for m , p and q in (16), we get

$$\begin{aligned} m &= -\frac{180}{\pi^4} \gamma(H) + \frac{30}{\pi^3} \phi(H) + \frac{180}{\pi^5} \zeta(H), \\ p &= \frac{192}{\pi^3} \gamma(H) - \frac{36}{\pi^2} \phi(H) - \frac{180}{\pi^4} \zeta(H), \\ q &= -\frac{36}{\pi^2} \gamma(H) + \frac{9}{\pi} \phi(H) + \frac{30}{\pi^3} \zeta(H). \end{aligned} \tag{20}$$

Table 2 shows the values of m , p , and q for several values of H . All these coefficients are calculated using 8192 terms in the expressions for $\phi(H)$, $\gamma(H)$ and $\zeta(H)$. We use this number of terms to be consistent with the error analysis described in Section 5.2.

In general, given a value of H , we initially use (17)–(19) to find the values of $\phi(H)$, $\gamma(H)$, and $\zeta(H)$, to finally calculate m , p and q using (20). Thus, $B_{2;\infty}(\lambda; H)$ can be approximated using $m\lambda^2 + p\lambda + q$ instead of the direct calculation of the infinity summation in (3) when $M = 2$ and $N \rightarrow \infty$. This indubitably makes this approach an excellent choice for long sequences where the computation of m , p and q is much less complicated than the computation of $B_{2;\infty}(\lambda; H)$ for each value of λ in U , where U is a set of u uniformly spaced values in $0 < \lambda \leq \pi$; i.e. $\lambda_1 = \pi/u$, $\lambda_2 = 2\pi/u$, $\lambda_3 = 3\pi/u, \dots, \lambda_u = \pi$.

4.2. Splines

For a given set of data, it is possible to use an n th-order polynomial to approximate this set of data in a limited range of values. However, the quality of the interpolation may not increase when the degree n of the polynomial is increased as established by Todd (1962) and Isaacson and Keller (1966). The method of splines effectively avoids this disadvantage and can be used to approximate (2). Splines were used for the first time in 1946 to substantially enhance the numerical stability of polynomial approximation, see Schoenbert (1946).

Spline interpolation is a piecewise polynomial interpolation, as indicated by Kreyszig (1993). To use splines to approximate $B(\lambda; H)$, we can rewrite (2) using (3) as follows:

$$\begin{aligned} B(\lambda; H) &= (2\pi + \lambda)^{-2H-1} + (2\pi - \lambda)^{-2H-1} + B_{2;\infty}(\lambda; H) \\ &= (2\pi - \lambda)^{-2H-1} + f(\lambda), \end{aligned}$$

where

$$f(\lambda) = (2\pi + \lambda)^{-2H-1} + B_{2;\infty}(\lambda; H). \tag{21}$$

Existing methods for approximating $B(\lambda; H)$ retain one, two or three terms of the infinite summation of (2)—see Paxson (1997), Ledesma and Liu (2000). Each term in this equation has two components, $(2\pi k - \lambda)^{-2H-1}$ and $(2\pi k + \lambda)^{-2H-1}$. Both components belong to the family of functions of the type x^y with $-3 \leq y \leq -2$ and $x \geq \pi$. These functions have a derivative of the form yx^{y-1} . For the same negative value of y , x^y changes more rapidly for small values of x than for large values of x . Specifically, for the same value of H , $(2\pi - \lambda)^{-2H-1}$ changes more rapidly than $(2\pi + \lambda)^{-2H-1}$ when λ changes. To reduce complexity and to maintain high accuracy, the spline method retains only the component with the highest variability, namely $(2\pi - \lambda)^{-2H-1}$. Consider the following equation as a method for approximating $B(\lambda; H)$:

$$B_{\text{Spline}}(\lambda; H) = (2\pi - \lambda)^{-2H-1} + g(\lambda), \tag{22}$$

where $B_{\text{Spline}}(\lambda; H)$ denotes the value of $B(\lambda; H)$ when the Spline method is used, and $g(\lambda)$ needs to be determined so that

$$g(\lambda) \approx f(\lambda),$$

in the interval $0 \leq \lambda \leq \pi$ and $0.5 \leq H < 1$. Following the standard definition of splines, the interval $0 \leq \lambda \leq \pi$ can be subdivided into r subintervals with common endpoints, called nodes, as

$$\lambda_0 < \lambda_1 < \lambda_2 < \dots < \lambda_r, \tag{23}$$

where $\lambda_0 = 0$ and $\lambda_r = \pi$. Now suppose that we require that $g(\lambda)$ and $f(\lambda)$ coincide at each of the endpoints, namely

$$g(\lambda_0) = f(\lambda_0), \quad g(\lambda_1) = f(\lambda_1), \dots, \quad g(\lambda_r) = f(\lambda_r), \tag{24}$$

and, that in each subinterval $g(\lambda)$ is given by an n th-order polynomial. Therefore, instead of approximating $f(\lambda)$ by a single polynomial on $0 \leq \lambda \leq \pi$, $f(\lambda)$ is approximated by r polynomials. This alternative procedure efficiently provides a more convenient method for approximating functions in some cases. Its official name is derived from thin elastic rods, called splines. For many years, these splines have been used to properly fit curves for data sets.

For each and every interval, as in (23), an n th-order polynomial can be used to approximate $f(\lambda)$. These polynomials can be functions of second or higher order. For practical purposes, cubic splines are very popular. By definition, a cubic spline must be a continuous function that has first and second derivatives everywhere in a specific interval as indicated by Kreyszig (1993). In each subinterval, $g(\lambda)$ is expressed by a polynomial of degree $n = 3$. Hence, $g(\lambda)$ has r polynomials,

$$g(\lambda) = \begin{cases} p_1(\lambda), & \lambda_0 < \lambda < \lambda_1 \\ p_2(\lambda), & \lambda_1 < \lambda < \lambda_2 \\ \vdots & \\ p_r(\lambda), & \lambda_{r-1} < \lambda < \lambda_r, \end{cases}$$

where

$$p_j(\lambda) = a_{j0} + a_{j1}(\lambda - \lambda_j) + a_{j2}(\lambda - \lambda_j)^2 + a_{j3}(\lambda - \lambda_j)^3,$$

for $j = 1, 2, \dots, r$, the coefficients a_{j0}, a_{j1}, a_{j2} and a_{j3} can be determined using Taylor's formula, see Kreyszig (1993). Turning our attention back to the approximation of $B(\lambda; H)$ in (22), we need to definitely find an appropriate value for r , the number of subintervals in $0 \leq \lambda \leq \pi$. This value of r must be an integer larger than 1. Otherwise, we will get only one polynomial for all values of the discrete frequency λ . A value of two or greater can be used; we will specifically show that a value of two impressively provides an acceptable accuracy. For $r = 2$ and equidistant nodes, the cubic spline method simplifies to

$$g(\lambda) = \begin{cases} a_{00} + a_{01}\lambda + a_{02}\lambda^2 + a_{03}\lambda^3, & 0 \leq \lambda < \pi/2, \\ a_{10} + a_{11}\left(\lambda - \frac{\pi}{2}\right) + a_{12}\left(\lambda - \frac{\pi}{2}\right)^2 + a_{13}\left(\lambda - \frac{\pi}{2}\right)^3, & \pi/2 \leq \lambda \leq \pi. \end{cases} \tag{25}$$

After using Taylor's formula and (24), we get the coefficients for $g(\lambda)$ for the interval $0 \leq \lambda < \pi/2$ as

$$\begin{aligned} a_{00} &= f(0), \\ a_{01} &= f'(0), \\ a_{02} &= \frac{12}{\pi^2} \left[f\left(\frac{\pi}{2}\right) - f(0) \right] - \frac{2}{\pi} [k_1 + 2f'(0)], \end{aligned}$$

and

$$a_{03} = \frac{16}{\pi^3} \left[f(0) - f\left(\frac{\pi}{2}\right) \right] + \frac{4}{\pi^2} [k_1 + f'(0)], \tag{26}$$

where

$$k_1 = \frac{3}{2\pi} [f(\pi) - f(0)] - \frac{1}{4} [f'(0) + f'(\pi)]. \tag{27}$$

For the remaining part of the interval $\pi/2 \leq \lambda \leq \pi$, the values of the polynomial coefficients are

$$\begin{aligned} a_{10} &= f\left(\frac{\pi}{2}\right), \\ a_{11} &= k_1, \\ a_{12} &= \frac{12}{\pi^2} \left[f(\pi) - f\left(\frac{\pi}{2}\right) \right] - \frac{2}{\pi} [f'(\pi) + 2k_1] \end{aligned}$$

and

$$a_{13} = \frac{16}{\pi^3} \left[f\left(\frac{\pi}{2}\right) - f(\pi) \right] + \frac{4}{\pi^2} [f'(\pi) + k_1]. \tag{28}$$

Note that the a_{ij} coefficients depend only on the value of $f(\lambda)$ for $\lambda = 0$, $\lambda = \pi/2$ and $\lambda = \pi$, and $f'(\lambda)$ for $\lambda = 0$ and $\lambda = \pi$. An expression of $f'(\lambda)$ can be found directly from (2) and (21) as

$$f'(\lambda) = (-2H - 1) \left[\sum_{k=1}^{\infty} (2\pi k + \lambda)^{-2H-2} - \sum_{k=2}^{\infty} (2\pi k - \lambda)^{-2H-2} \right]. \tag{29}$$

Tables 3 and 4 show the values of the spline method coefficients for several values of H . All these coefficients were calculated using 8192 terms in the expressions for $f(H)$ and $f'(H)$. Note that we use this number of terms to be consistent with the error analysis described in Section 5.2.

Table 3
Spline method coefficients for several values of H and $0 \leq \lambda < \pi/2$

H	a_{00}	a_{01}	a_{02}	a_{03}
0.6	3.4746×10^{-2}	-6.1411×10^{-3}	1.6618×10^{-3}	-1.8217×10^{-4}
0.7	2.1455×10^{-2}	-4.6387×10^{-3}	1.3011×10^{-3}	-1.5016×10^{-4}
0.8	1.3546×10^{-2}	-3.4796×10^{-3}	1.0107×10^{-3}	-1.2222×10^{-4}
0.9	8.6989×10^{-3}	-2.5946×10^{-3}	7.7989×10^{-4}	-9.8377×10^{-5}

Table 4
Spline method coefficients for several values of H and $\pi/2 \leq \lambda \leq \pi$

H	a_{10}	a_{11}	a_{12}	a_{13}
0.6	2.8493×10^{-2}	-2.2690×10^{-3}	8.0330×10^{-4}	-3.4403×10^{-5}
0.7	1.6797×10^{-2}	-1.6629×10^{-3}	5.9344×10^{-4}	-2.7220×10^{-5}
0.8	1.0101×10^{-2}	-1.2089×10^{-3}	4.3482×10^{-4}	-2.1227×10^{-5}
0.9	6.1663×10^{-3}	-8.7274×10^{-4}	3.1630×10^{-4}	-1.6339×10^{-5}

Summarizing, for a given value of H , we calculate $f(0)$, $f(\pi/2)$ and $f(\pi)$ using Eq. (21). Then, we subsequently calculate $f'(0)$ and $f'(\pi)$ using Eq. (29). That is, we need to evaluate five infinite summations to use the Spline method instead of the evaluation of Eq. (2) for each value of λ in U . Note that the number of elements of the set U depends directly on the sequence length.

Once the values of $f(0)$, $f(\pi/2)$, $f(\pi)$, $f'(0)$ and $f'(\pi)$ have been determined, we use Eq. (27) to find the value of k_1 . Finally, Eqs. (26) and (28) can be used to compute the a_{ij} coefficients. These coefficients can be directly used in Eq. (25) to find an expression for $g(\lambda)$, which can be used, in turn, to approximate $B(\lambda; H)$ utilizing Eq. (22).

5. Analysis and simulation results

We proceed to analyze the computational complexity and accuracy of the methods for synthesizing the power spectrum of FGN. The section ends with an analysis to estimate the Hurst parameter using Whittle's estimator through the approximation methods of Sections 3 and 4.

5.1. Computational complexity

In order to sensibly compare the computational complexity of the different methods for approximating (2), we created 1000 series with $H = 0.7$ using all approximation methods, including FOA, Paxson, EM, SOA and Spline. Table 5 shows the average running time in microseconds of each of the methods for approximating (2) using a Pentium(R) 4 CPU 3.2 GHz. Table 6 shows the relative running time utilizing the FOA method as reference.

From Table 6, it can be seen that the EM and FOA methods have similar running time. Ordering from faster to slower, we have the Spline method at the top. Right after that, we have the SOA method, which is twice as fast as the FOA and the EM methods. Next, we proceed to evaluate the accuracy of all these approximations.

5.2. Approximation error

Before we compare the error introduced by the approximation methods discussed, we present a crucial discussion about how many terms in Eq. (2) are required to keep the error below a desired value.

Obviously, if more terms are included in the summation of Eq. (2), the computed value of $B(\lambda; H)$ will be closer to the exact value. However, to correctly assess the gradual effect of cutting off the last terms of the summation in (2), it is important to consider the value of N in Eq. (3). In practice, N must be chosen to be big enough to attain a specific desired error. However, $B(\lambda; H)$ decreases slowly and each term has a significant contribution to the total value of the summation. Therefore, instead of finding the ratio of the last term over the value of the total summation, we compare the summation of all the last terms with the value of the total summation. Thus, we define

$$\text{Error}(n, N) = \frac{B_{n:N}(\lambda; H)}{B_{1:N}(\lambda; H)}, \quad N \gg n, \tag{30}$$

as the error introduced for taking into consideration only the n -first terms of (2). Table 7 shows the value of $\text{Error}(n, 10^5)$ for $\lambda = \pi/2$. From this table, we see that for values of H close to 1, fewer terms are required than for values of H close to 0.5.

Evaluating $\text{Error}(n, N)$, for different values of $n \ll N$, is possible to find the number of terms required to attain a specific error. In this paper and for simulation purposes, the exact value of $B(\lambda; H)$ was calculated using 8192 terms. For a value of $H = 0.95$, using 8192 terms provides an error of 1.1597×10^{-8} ; while for $H = 0.55$ the same number of terms provides 2.3394×10^{-5} .

Now, we proceed to compare all these methods using the mean absolute error, specifically defined as

$$\text{MAE}(\lambda; H) = \frac{1}{u} \sum_{n=0}^{u-1} |B_{\text{approx}}(\lambda_n; H) - B_{\text{exact}}(\lambda_n; H)|,$$

where u is the number of elements in the set U . Fig. 1 shows the mean absolute error as a function of H using 8192 values of λ ($u = 8192$), and 1024 values for H . It can be quickly observed that all methods present a better approximation for values of H close to 1. The minimum mean absolute error is attained by the Spline method. The SOA method and

Table 5
Running time in microseconds (average of 1000 series with $H = 0.7$)

Length	Paxson	FOA	SOA	EM	Spline
8192	8747	3514	1820	3549	988
16,384	17,674	7120	3673	7159	1971
32,768	34,935	14,083	7222	14,094	3961
65,536	72,169	28,095	14,388	28,170	7925

Table 6
Relative running time with respect to FOA (average of 1000 series with $H = 0.7$)

Length	Paxson	FOA	SOA	EM	Spline
8192	2.4889	1.0000	0.57784	1.0098	0.28111
16,384	2.4822	1.0000	0.51580	1.0055	0.27681
32,768	2.4809	1.0000	0.51287	1.0009	0.28126
65,536	2.5688	1.0000	0.51209	1.0027	0.28207

Table 7
Error introduced for using only the first n terms for $B(\lambda; H)$

n	$H = 0.55$	$H = 0.95$
4096	5.1972×10^{-5}	4.3560×10^{-8}
8192	2.3394×10^{-5}	1.1597×10^{-8}
16,384	1.0063×10^{-5}	3.0332×10^{-9}

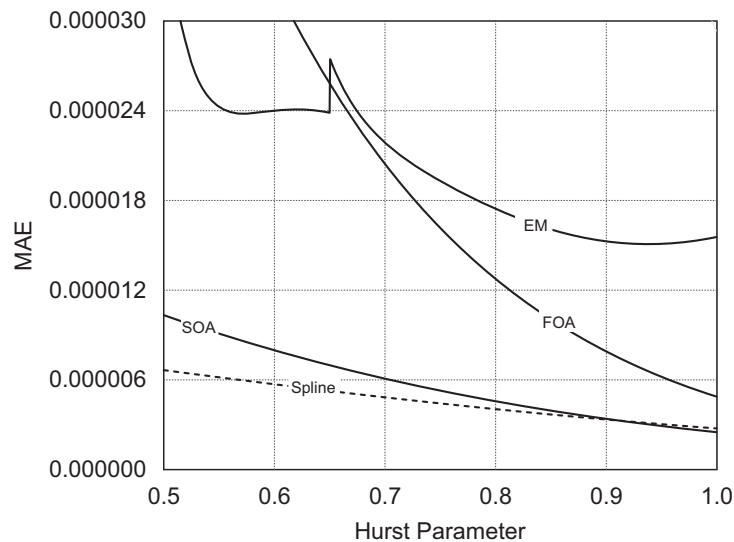


Fig. 1. Mean absolute error for different values of H .

the Spline method provide similar mean absolute errors for values of H bigger than 0.83. The FOA method exhibit a smaller mean absolute error than the EM method for H bigger than 0.65. On the other hand, the EM method offers a better accuracy than the FOA method for values of $H < 0.65$. Note the irregular behavior of the EM method due to the use of two sets of c -coefficients as clearly indicated in Section 3.

Table 8
Relative mean absolute error using the FOA method as reference

H	Paxson	FOA	SOA	EM	Spline
0.6	7.6123	1.0000	0.24524	0.73740	0.17535
0.7	7.1595	1.0000	0.29735	1.0696	0.23638
0.8	6.7478	1.0000	0.35845	1.3682	0.31720
0.9	6.3641	1.0000	0.42986	1.9317	0.42397

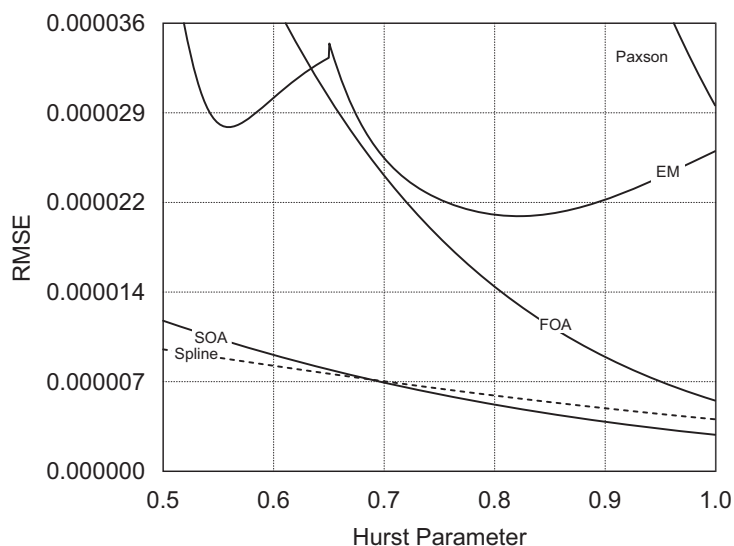


Fig. 2. Root mean-squared error as a function of H for all approximation methods.

Table 8 shows the relative mean absolute error for different values of H . In this table, the FOA method has been used as reference. Thus, its relative mean absolute error is 1 for all H . From this table, it can be seen that Paxson method has an error, which is six to eight times greater than the FOA method. The SOA method has an error that is at least three times smaller than the FOA method. Lastly, the Spline method has an error that is two to six times smaller than the FOA method.

Because of its accuracy and the fact that the Spline method is the fastest, we ultimately conclude that the Spline method is an excellent choice for approximating the power spectrum of FGN.

Next, we use the root mean-squared error to compare the accuracy of the present methods with existing approximation methods. We start by defining the root mean-squared error as

$$RMSE(\lambda; H) = \sqrt{\frac{1}{u} \sum_{n=0}^{u-1} [B_{\text{approx}}(\lambda_n; H) - B_{\text{exact}}(\lambda_n; H)]^2},$$

where u , as before, is the number of values of λ in the set U . Fig. 2 shows the rmse for $u = 8192$ and 1024 values for H ; it can be seen that the SOA and the Spline methods have a much smaller rmse than other methods to approximate $B(\lambda; H)$. Fig. 3 shows the Theil U inequality index for all the approximation methods—see Theil (1972). From this figure, it can be seen that the Theil U inequality index and the rmse provide consistent results. Table 9 shows the relative root mean-squared error for different values of the Hurst parameter.

5.3. LRD sequence synthesis and estimation of the Hurst parameter

We generated 1000 LRD sequences of length 8192 using the SOA method for several values of H . Then, we estimate the Hurst parameter using the variance–time plot, the periodogram and Whittle’s estimator. Tables 10 and 11 show the mean, variance, bias and range of the Hurst parameter for the Spline and SOA methods, respectively.

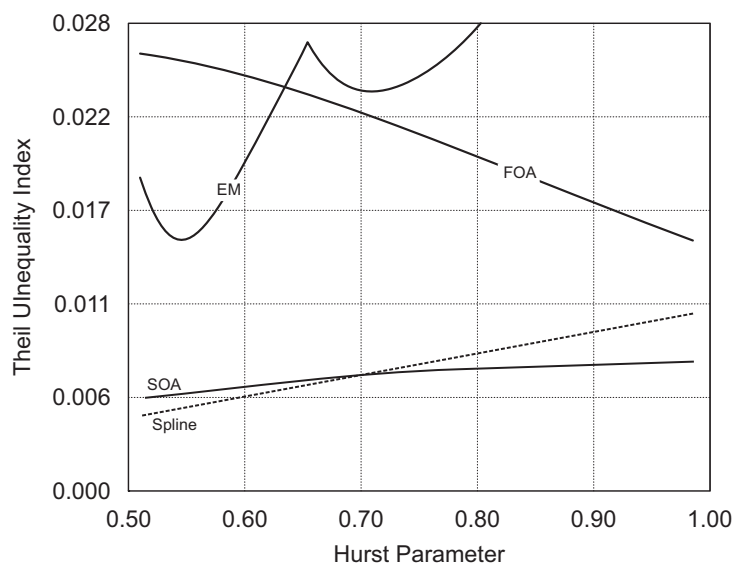


Fig. 3. Theil U inequality index as a function of H for all approximation methods.

Table 9
Relative root mean-squared error using as reference the FOA method

H	Paxson	FOA	SOA	EM	Spline
0.60000	6.5522	1.0000	0.2470	0.7932	0.2240
0.70000	6.1632	1.0000	0.2996	1.0585	0.3041
0.80000	5.8096	1.0000	0.3612	1.3908	0.4105
0.90000	5.4802	1.0000	0.4332	2.3749	0.5511

Table 10
Mean, variance, bias and range of the Hurst parameter for 1000 synthetic LRD sequences (Spline)

	H	Mean	Variance	Bias	Range
Variance–time plot	0.6	0.5948	0.0004	0.0052	[0.5272, 0.6404]
	0.7	0.6842	0.0005	0.0158	[0.6340, 0.7490]
	0.8	0.7773	0.0006	0.0227	[0.7094, 0.8370]
	0.9	0.8506	0.0006	0.0494	[0.7874, 0.9058]
Periodogram	0.6	0.6028	0.0004	−0.0028	[0.5540, 0.6605]
	0.7	0.7055	0.0002	−0.0055	[0.6760, 0.7451]
	0.8	0.8075	0.0003	−0.0075	[0.7567, 0.8580]
	0.9	0.9093	0.0003	−0.0093	[0.8648, 0.9515]
Whittle’s estimator	0.6	0.6007	0.0000	−0.0003	[0.5840, 0.6160]
	0.7	0.6999	0.0000	0.0001	[0.6864, 0.7134]
	0.8	0.8010	0.0001	−0.0002	[0.7754, 0.8168]
	0.9	0.9004	0.0000	−0.0004	[0.8851, 0.9188]

5.4. Whittle’s estimator

A related problem when analyzing samples of self-similar processes is estimating the Hurst parameter H . As previously mentioned, Whittle’s estimator can be used to calculate H based on a maximum likelihood technique. Thus, the approximation methods reviewed in this paper can be used to quickly evaluate Whittle’s estimator, see Beran (1992b).

Table 11
Mean, variance, bias and range of the Hurst parameter for 1000 synthetic LRD sequences (SOA)

	H	Mean	Variance	Bias	Range
Variance–time plot	0.6	0.5977	0.0004	0.0023	[0.5411, 0.6441]
	0.7	0.6892	0.0005	0.0108	[0.6284, 0.7390]
	0.8	0.7731	0.0005	0.0269	[0.7273, 0.8328]
	0.9	0.8468	0.0004	0.0532	[0.7912, 0.8919]
Periodogram	0.6	0.6054	0.0004	−0.0054	[0.5578, 0.6468]
	0.7	0.7067	0.0003	−0.0067	[0.6612, 0.7492]
	0.8	0.8058	0.0003	−0.0058	[0.7480, 0.8463]
	0.9	0.9090	0.0003	−0.0090	[0.8801, 0.9550]
Whittle's estimator	0.6	0.6004	0.0001	−0.0004	[0.5797, 0.6188]
	0.7	0.7007	0.0001	−0.0007	[0.6846, 0.7170]
	0.8	0.8002	0.0000	−0.0002	[0.7862, 0.8155]
	0.9	0.9005	0.0001	−0.0005	[0.8870, 0.9247]

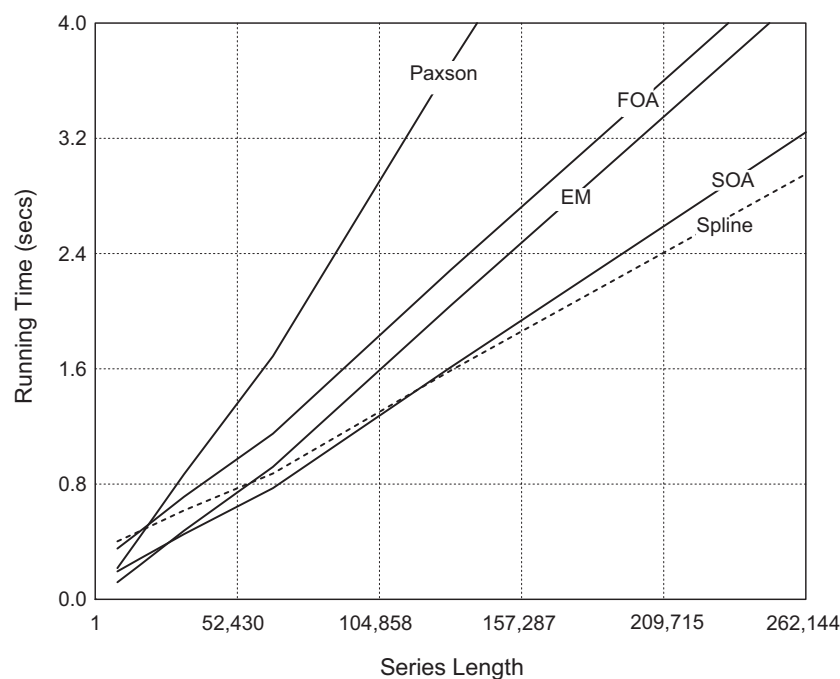


Fig. 4. Whittle's estimator running time for all approximation methods.

Fig. 4 shows the running time to compute Whittle's estimator using these approximation methods using a Pentium(R) 4 CPU 3.2 GHz. As it can be seen from this figure, the running time grows approximately linearly for all methods. Additionally, it can be seen that for short sequences most of the methods perform similarly. However, for long sequences, the Spline method is the fastest.

6. Summary and future work

There are several methods for approximating the power spectrum of FGN. Paxson (1997) proposed a method using the midpoint of two integrals. Ledesma and Liu (2000) proposed the first-order approximation (FOA) method to significantly reduce the computational complexity of Eq. (2) while keeping high accuracy. Recently, another method, based on the Euler–Maclaurin summation, was proposed by Purczynsky and Wlodarski (2006). All these approximation methods have considerably reduced the direct calculation of the infinite summation of Eq. (2), making extremely difficult to propose new methods to compete with existing ones.

In this paper, we developed two new methods for approximating the power spectrum of FGN. The first method, called second-order approximation (SOA), indirectly uses a second-order function to approximate the power spectrum of FGN. This new method is 2 times faster than the FOA method presented by Ledesma and Liu (2000). The second method, called the Spline method, is 3.6 times faster than the FOA method. Even though splines cannot be used directly to approximate Eq. (2), they can simplify the calculation without sacrificing exactitude.

We showed that the SOA and Spline methods are faster, and have a better accuracy than existing methods. For those cases where speed is mandatory, i.e., Whittle’s estimator, the Spline method is a better choice than the SOA method. We adequately provide a statistical analysis of existing methods and the two present ones. We use the mean absolute error and the root mean-squared error to compare the accuracy of all these methods. We adequately provide an analysis to sensibly compare the computational complexity of these approaches. Finally, we performed computer simulation studies to verify all our results.

Future research includes the viability to extend these methods to the non stationary long memory case.

Acknowledgment

This work was supported by the National Science Foundation under Grants ANI-0203063 and PROMEP.

Appendix A. The SOA method for $H = 0.5$

LRD sequences are those with $0.5 < H < 1$. Conversely, short range-dependent sequences exhibit a value of $H = 0.5$. Eqs. (18) and (19) cover all the values of H required for LRD sequence synthesis. However, it is customary to get an expression for short range dependence. Using the previous definition for α_k , β_k , and ρ_k , the expressions of $\gamma(H)$ and $\zeta(H)$ for $H = 0.5$ are given by

$$\gamma(H) = \sum_{k=2}^{\infty} \left[\ln \frac{\alpha_k^2 - 1}{\alpha_k^2} + \frac{2}{\alpha_k^2 - 1} \right] \tag{A.1}$$

and

$$\zeta(H) = 2\pi \sum_{k=2}^{\infty} \left[1 + \frac{\alpha_k^2}{\alpha_k^2 - 1} + \alpha_k \ln \frac{\beta_k}{\rho_k} \right]. \tag{A.2}$$

Appendix B. Convergence of the SOA method series

The convergence of $\phi(H)$, $\gamma(H)$ and $\zeta(H)$ of Eqs. (17)–(19) will be briefly discussed. After expanding Eq. (17) and some algebra, this equation reduces to

$$\phi(H) = \frac{\pi^{-2H}}{2H} \left[3^{-2H} - \lim_{k \rightarrow \infty} \frac{1}{k^{2H}} \right] = \frac{(3\pi)^{-2H}}{2H},$$

thus, $\phi(H)$ converges for any $0.5 \leq H < 1$.

To prove that $\gamma(H)$ and $\zeta(H)$ are convergent functions, we write Eqs. (18) and (19) as

$$\gamma(H) = \sum_{k=2}^{\infty} g(H; k) \quad \text{and} \quad \zeta(H) = \sum_{k=2}^{\infty} z(H; k),$$

where $g(H; k)$ and $z(H; k)$ are positive monotone decreasing functions of k defined on the interval $[2, \infty)$. After some algebra, it is possible to prove that

$$\int_2^{\infty} g(H; k) dk = \frac{\pi^\mu}{\eta\mu} \left[\frac{3^\mu - 5^\mu}{2} - \frac{4^{\eta+2}}{\eta + 2} + \frac{3^{\eta+2} + 5^{\eta+2}}{2(\eta + 2)} \right] < \infty$$

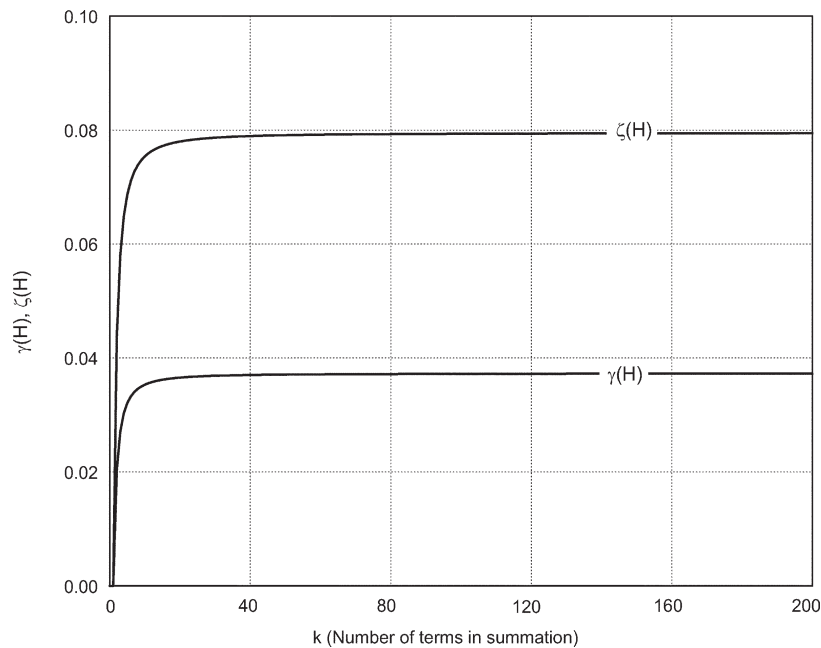


Fig. B1. Values of $\gamma(H)$ and $\zeta(H)$ as a function of k for $H = 0.75$.

and

$$\int_2^{\infty} z(H; k) dk = \frac{\pi^{\eta+2}}{\eta\mu} \left[\frac{3^{\eta+1}(\eta^2 + 11\eta + 42) - 5^{\eta+1}(\eta^2 - 5\eta + 26)}{2(\eta + 3)(\eta + 2)} \right] < \infty.$$

Thus, using the convergence integral test, it is concluded that the series for $\gamma(H)$ and $\zeta(H)$ of Eqs. (18) and (19) converge for any $0.5 < H < 1$. Fig. B1 shows the values of $\phi(H)$, $\gamma(H)$ and $\zeta(H)$ as a function of the number of terms in the summation of Eqs. (18) and (19) for $H = 0.75$. From this figure, it can be seen that these functions converge rapidly, and only few terms are required to get an excellent approximation.

Appendix C. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csda.2007.02.017.

References

- Adas, A., Mukherjee, A., 1995. On resource management and QoS guarantees for long range dependent traffic. In: Proceedings of the IEEE Conference on Computer Communications INFOCOM 2, 779–787.
- Beran, J., 1986. Estimation, testing and prediction for self-similar and related processes. Ph.D. Dissertation, Swiss Federal Institute of Technology, Zurich, Switzerland.
- Beran, J., 1992a. A goodness-of-fit test for time series with long range dependence. *J. Roy. Statist. Soc.* 54, 749–760.
- Beran, J., 1992b. Statistical methods for data with long-range dependence. *Statist. Sci.* 7, 404–427.
- Beran, J., 1994. *Statistics for Long-Memory Processes*. Chapman & Hall, New York.
- Beran, J., Sherman, R., Taqqu, M.S., Willinger, W., 1995. Long-range dependence in variable-bit-rate video traffic. *IEEE Trans. Commun.* 43, 1566–1579.
- Cox, D.R., 1984. Long-range dependence: a review. Proceedings of the 50th Anniversary Conference, H.T. David Iowa State University, 55–74.
- Dahlquist, G., Bjorck, A., 1974. *Numerical Methods*. Prentice-Hall, Upper Saddle River, NJ.
- Erramilli, A., Gordon, J., Willinger, W., 1994. Applications of fractals in engineering for realistic traffic processes. In: Proceedings of ITC-14. Antibes Juan-les-Pins, France, pp. 35–44.
- Erramilli, A., Narayan, O., Willinger, W., 1996. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. Networking* 4, 209–223.
- Flandrin, P., 1992. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Trans. Inform. Theory* 38, 910–917.
- Fox, R., Taqqu, M., 1986. Large-sample properties of parameter estimates for strongly dependent stationary Gaussian time series. *Ann. Statist.* 14, 517–532.

- Isaacson, E., Keller, H., 1966. *Analysis of Numerical Methods*. Wiley, New York.
- Jeong, H.D., 2002. *Modelling of self-similar teletraffic for simulation*. Ph.D. Thesis, Department of Computer Science, University of Canterbury, New Zealand.
- Kreyszig, E., 1993. *Advanced Engineering Mathematics*. seventh ed. Wiley, New York. pp. 949–956.
- Ledesma, S., Liu, D., 2000. Synthesis of fractional Gaussian noise using linear approximation for generating self-similar network traffic. *ACM Comput. Commun. Rev.* 30, 4–17.
- Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V., 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Networking* 2, 1–15.
- Paxson, V., 1997. Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic. *ACM SIGCOMM Comput. Commun. Rev.* 27, 5–18.
- Paxson, V., Floyd, S., 1995. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Networking* 3, 226–244.
- Purczynski, J., Wlodarski, P., 2006. On fast generation of fractional Gaussian noise. *Comput. Statist. Data Anal.* 50 (10), 2537–2551.
- Roughan, M., Veitch, D., Abry, P., 1998. On-line estimation of the parameters of long-range dependence. In: *Proceedings of GLOBECOM'98*, Sydney, Australia, 3716–3721.
- Schoenbert, I.J., 1946. *Quarterly of applied mathematics*. *Appl. Math.* 4, 45–99.
- Spiegel, M.R., Liu, J., 1998. *Schaum's Mathematical Handbook of Formulas and Tables*. second ed. McGraw-Hill, New York, NY.
- Taqqu, M.S., Teverovsky, V., 1997. Is network traffic self-similar or multifractal? *Fractals* 5, 66–73.
- Theil, H., 1972. *Statistical Decomposition Analysis*. North-Holland Publishing Company, Amsterdam.
- Todd, J., 1962. *Survey of Numerical Analysis*. McGraw-Hill, New York.
- Veitch, D., Abry, P., 1998. Wavelet analysis of long-range dependent traffic. *IEEE Trans. Inform. Theory* 44, 2–15.
- Veitch, D., Abry, P., 1999. A wavelet based joint estimator of the parameters of long-range dependence. *IEEE Trans. Inform. Theory* 45, 878–897.
- Willinger, W., Taqqu, M.S., Leland, W.E., Wilson, D.V., 1995. Self-similarity in high-speed packet traffic: analysis and modeling of Ethernet traffic measurement. *Statist. Sci.* 10, 67–85.
- Willinger, W., Taqqu, M.S., Sherman, R., Wilson, D.V., 1997. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Trans. Networking* 5, 71–86.